FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Das ZAM-ATM-Testbett
Erfahrungen beim Betrieb
Ergebnisse von Durchsatzmessungen

Adam Lukosek, Martin Sczimarowsky,
Sabine Werner

KFA-ZAM-IB-9634

Dezember 1996
(Stand 04.12.96)

Inhaltsverzeichnis

Verzeichnis der Tabellen

Verzeichnis der Abbildungen

1 Einleitung

Im März 1995 wurde das ATM-Netz des Regionalen Testbeds (RTB) NRW mit den angeschlossenen Einrichtungen RWTH Aachen, Universität zu Köln, DLR Köln, GMD St. Augustin, Universität Bonn und Forschungszentrum Jülich (KFA) in Betrieb genommen. Als normaler Zugang zu diesem Netz stand eine Router-Schnittstelle zur Verfügung, für Testzwecke konnten über ein zusätzliches Switch-Interface lokale ATM-Netze angeschlossen werden.

Zeitgleich damit wurde im Zentralinstitut für Angewandte Mathematik (ZAM) der KFA ein ATM-Testbett aufgebaut, um den direkten ATM-Zugang zum RTB zu eröffnen, erste Erfahrungen mit Produkten und den von diesen Produkten unterstützten ATM-Standards zu sammeln und so zu einer Einschätzung der Einsetzbarkeit von ATM - insbesondere im KFAnet - unter den derzeitigen Randbedingungen zu gelangen.

Die eingesetzte Hardware im ATM-Testbett des ZAM stammt aus später zu erläuternden Gründen von verschiedenen Herstellern. Diese Tatsache und die unterschiedliche Unterstützung der ATM-Standards führten im Verlauf der Arbeiten an vielen Stellen zu Problemen.

In diesem Bericht werden die Erfahrungen bei der Installation und beim Betrieb des Testbetts in der KFA geschildert und die Ergebnisse von Durchsatzmessungen dargestellt. Auf eine Darstellung der Grundlagen der ATM-Technik wird weitgehend verzichtet (siehe hierzu z.B. [3]).

2 Das ZAM-ATM-Testbett

2.1 Verkabelung

Das ZAM-ATM-Testbett wurde komplett auf der Basis von Glasfaser konzipiert. Die Voraussetzungen hierfür wurden um den Jahreswechsel 1994/1995 mit der Installation einer sternförmigen Glasfaser-Gebäudeverkabelung (62.5 / 125μm Multimode) geschaffen.

2.2 ATM-Switches

Für das Testbett wurden zwei ATM-Switches der Firma Cisco vom Typ LightStream 100 beschafft. Der LightStream 100 ist eine Entwicklung der Firma NEC und wurde ursprünglich unter dem Namen HyperSwitch A100 vertrieben (2.4 Gbps Durchsatz, non blocking, input/output buffer type switch fabric).

Folgende Gründe beeinflußten die Entscheidung für diesen Switch:

- Interoperabilität mit den im RTB eingesetzten Switches vom selben Typ
- Interoperabilität mit den im ZAM eingesetzten Cisco-Routern
- vergleichsweise niedriger Preis der Switches im Rahmen einer Sonderaktion der Firma Cisco

Jeder Switch ist mit 7 Interfaces STS3c/STM1 Fiber (155 Mbps) sowie 3 Interfaces TAXI 4B/5B (100 Mbps) ausgestattet. Die Geräte wurden mit dem Software-Release 1.2(0) ausgeliefert und unterstützten damit laut Dokumentation folgende ATM-Standards:

- User Network Interface (UNI) V3.0
- ITU-T Q.93B
- Private Network Node Interface (P-NNI, nicht weiter spezifiziert)

In der Folge wurden Upgrades auf die Releases 2.3(5) (im Wesentlichen erweiterte Konfigurationskommandos), 2.5(2) (Soft PVC/PVP, IISP) und schließlich 3.2(5) durchgeführt. Dieses Release hat zusätzlich folgende Features:

- ILMI (Interim Local Management Interface: Setzen der VPI/VCI-bits in Abstimmung zwischen Switch und Endgerät, Adreßregistrierung, LECS-Adresse)
- LANE (LAN-Emulation: nur LECS-Adresse (siehe ILMI))

2.3 Router

Zur Anbindung des ATM-Testbett

s an das KFAnet wurde ein Cisco-Router vom Typ Cisco 7000 mit zwei ATM-155 Mbps Interfaces und einem FDDI-Interface eingesetzt. Die Software entsprach dem zum jeweiligen Zeitpunkt aktuellen 11.0(x) Maintenance Release.

Der Cisco 7000 unterstützt mit der Softwareversion 11.0 folgende ATM-Standards und -Serverfunktionen:

- Classical IP: ATMARP Client/Server

- LANE:
  - LAN Emulation Server (LES)
  - LAN Emulation Client (LEC)
  - LAN Emulation Configuration Server (LECS)
  - Broadcast and Unknown Server (BUS)
- ILMI
- UNI 3.0

2.4 ATM-Endgeräte

2.4.1 Sun

Bei den ATM-Tests wurde zwei Sun SPARCstation 20 unter Solaris 2.4 eingesetzt. Beide wurden mit SunATM-155/MFiber SBus-Adaptern (Version 1.0) ausgestattet. Die mit dem Adapter 1.0 gelieferte Version 1.0 der ATM-Software unterstützt folgende ATM-Standards:

- ATM Adaption Layer (AAL) 5
- UNI 3.0
- Classical IP over ATM (RFC 1577), SVC/PVC, ATMARP-Client mit dynamischer Adreßauflösung oder Adreßauflösung anhand statischer lokaler Tabellen, ATMARP-Server
- ILMI

Im April 1996 wurde die Software durch die neue Version 2.0 ersetzt, die für den neuen 2.0–Adapter entwickelt wurde, aber auch den alten Adapter unterstützt. Diese Software brachte als wesentliche Neuerung die Unterstützung eines LANE-Clients.

Eine der beiden SPARCstation 20 wurde im Juli 1996 durch eine SPARCstation 5 ersetzt.

2.4.2 DEC

Zuerst wurden zwei DEC 3000 M 300 mit je einem Turbochannel DEC*ATMworks 750* 155 Mbps Adapter in des Testnetz eingebunden. Später wurden eine AlphaStation 200 4/100 und ein AlphaServer 1000 4/266 mit je einem PCI Bus DEC *ATMworks 350* 155 Mbps Adapter ausgestattet und in Betrieb genommen. Die Unterstützung für den PCI Bus Adapter erfolgt ab der Betriebssystem-Version Digital UNIX V3.2c.

Die Betriebssystem-Versionen DEC OSF/1 V3.2a, V3.2b und Digital UNIX V3.2c unterstützen

- AAL 5
- UNI 3.0
- *Classical IP over ATM* nach RFC 1577
- ILMI

Mit Digital UNIX 4.0 wird der ATM-Funktionalitäts-Umfang um LANE in Form des LECs erweitert.

2.4.3 IBM

Zwei IBM RS/6000-41T wurden mit 100 Mbps Turboways 100 TAXI ATM-Adaptern ausgestattet, während des Testbetriebs wurde auf beiden Maschinen ein Betriebssystem-Upgrade von AIX 3.2.5 nach AIX 4.0.1 durchgeführt. Die Maschinen haben bezüglich ATM folgende Eigenschaften:

- UNI 3.0
- AAL 5
- Classical IP: PVC/SVC, ATMARP Server/Client ab AIX 4.0.1

2.5 ATM-Protokoll-Tester

Zur Analyse von Problemen im ATM-Netz wurde ein ATM-Protokolltester vom Typ Telenex Interview 8750 für ATM Emulation, Tracing und Statistik beschafft. Der Tester verfügt über optische SONET/OC-3c (STS3c/STM1) – sowie E3–Schnittstellen. Er unterstützt ATM Monitor- und Statistikfunktionen, die Emulation der Netzwerk- wie der User-Seite eines Links und erlaubt Protokoll-Analysen auf verschiedenen Ebenen. Unterstützt werden z.B. UNI, RFC1577, LANE, ILMI, RFC1483.

3 Die Konfiguration des ZAM-ATM-Testbetts

Die folgenden Abbildungen zeigen die physikalische Konfiguration des ZAM-ATM-Testbetts für Classical IP und LAN Emulation, sowie die Anbindung an das KFAnet bzw. das RTB-NRW:

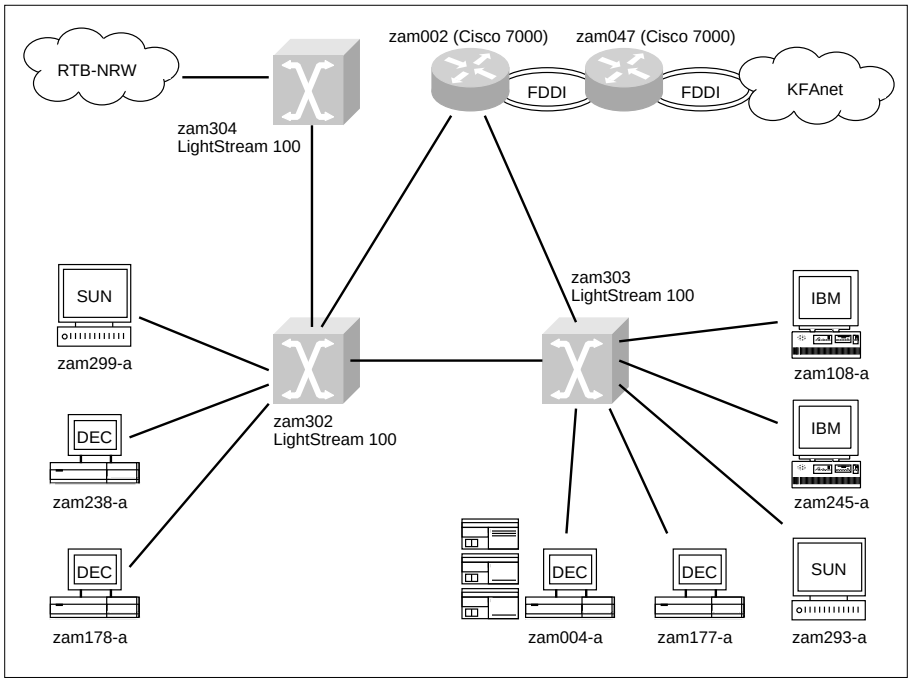

Abb. 1: Konfiguration für Classical IP over ATM

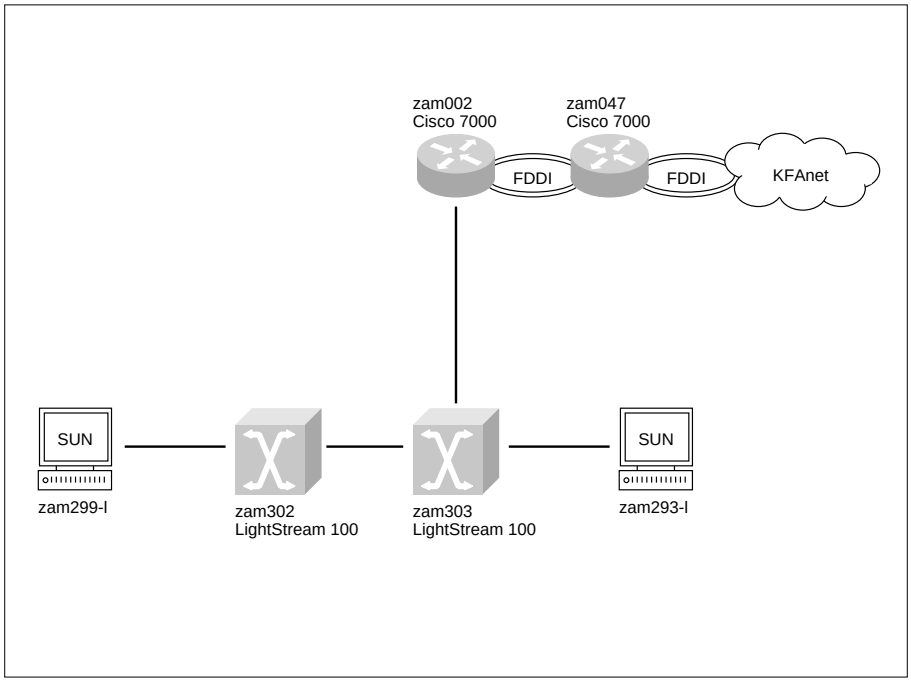

Abb. 2: Konfiguration für LAN Emulation

4 Das ATM-Labor im ZEL

Im April 1996 fand im ZEL eine ATM-Schulung durch die Firma DEC statt, bei der auch ATM-Equipment des ZAM eingesetzt wurde (je eine DEC- und Sun-Workstation, ein LS100 ATM-Switch sowie der Protokoll-Tester). Daneben wurden in dem Labor-Aufbau ein ATM-Switch vom Typ Gigaswitch der Firma DEC, ein DEC-Switch 400 sowie einige ATM-Endgeräte (Sun SPARCstation 5 mit SUN SBus-Karte, SGI mit FORE-ATM-Karte, eine weitere DEC-Maschine und ein NT-PC) eingesetzt.

In dieser gegenüber dem ZAM-ATM-Testbett noch vielfältigeren Umgebung traten erwartungsgemäß an vielen Stellen Interoperabilitätsprobleme auf. Ein Teil dieser Probleme konnte mit großem Aufwand und sehr tiefgehenden Kenntnissen der ATM-Technik (der Kursleiter arbeitet an Projekten des ATM-Forums mit) behoben oder umgangen werden. Andere Probleme mußten wegen der zeitlichen Rahmenbedingungen offengelassen oder durch Restart der betroffenen Komponenten beseitigt werden. Der Protokoll-Tester erwies sich bei der Problemanalyse als äußerst wertvoll.

Durch Einsatz des DEC-Switch 400 konnte hier auch die im LANE-Standard vorgesehene direkte Anbindung herkömmlicher LANs (in diesem Falle: Ethernet) an ein ATM-Netz getestet werden.

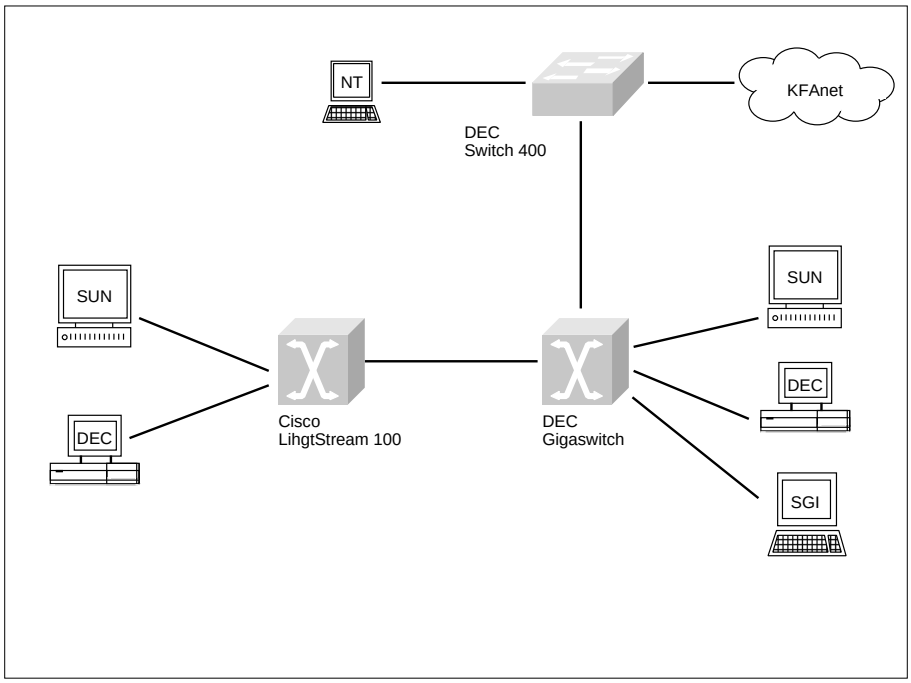

Abb. 3: Das ATM-Labor im ZEL

5 Betrieb: Classical IP / LAN Emulation (LANE)

Glaubt man den Äußerungen von Mitarbeitern der Hersteller bei der Präsentation von ATM-Produkten, so ist der Aufbau und die Konfiguration eines ATM-Netzes denkbar einfach. Nach der Festlegung der Funktion eines Endgerätes im ATM-Kontext (z.B. ATMARP-Client/Server) und der Festlegung von ATM-Adreßinformation auf den Switches registrieren sich die Endgeräte unter Verwendung des ILMI-Protokolls beim Switch und gegebenenfalls als ATMARP-Clients beim zugehörigen Server, und es können TCP/IP-Verbindungen über ATM aufgebaut werden.

Dies mag stimmen, wenn ausschließlich Komponenten eines einzigen Herstellers eingesetzt werden. Die Realität in einem ATM-Netz mit heterogener Hardware wie dem beschriebenen ATM-Testbett des ZAM sieht anders aus.

Viele wichtige Standards im ATM-Umfeld befinden sich zur Zeit noch in unterschiedlichen Entwicklungsstadien, was dazu führt, daß sich der Implementierungsstand von Hersteller zu Hersteller unterscheidet. Während des Testzeitraums waren zum Teil mehrere Software- und sogar Hardware-Upgrades erforderlich. Die Tatsache, daß wichtige Standards noch nicht verfügbar sind, führte darüberhinaus dazu, daß Hersteller für wichtige Funktionen proprietäre Lösungen implementieren, die nur zwischen den ATM-Komponenten dieses Herstellers eingesetzt werden können.

Eine weitere Ursache für Probleme lag in der Qualität der Benutzerführung. Die mit den ATM-Produkten gelieferte Dokumentation ist oft unvollständig und in den vorhandenen Teilen unzureichend. Dies führte zumindest in der Anfangsphase der Tests zu Unsicherheiten und Mißverständnissen.

5.1 VPI/VCI

ATM-Switching erfolgt auf der Basis eines Virtual Path Identifiers (VPI) und eines Virtual Circuit Identifiers (VCI), durch die auf einem Link im ATM-Netz eine Verbindung eindeutig identifiziert ist.

In der Definition des Headers einer ATM-UNI-Zelle sind 8 Bits für die Aufnahme des VPI und 16 Bits für die Aufnahme des VCI vorgesehen. Diese beiden Felder werden von der im Testbett eingesetzten Hardware und Software in sehr unterschiedlicher Weise unterstützt. Insbesondere benutzen die Endgeräte nur VPI=0. DEC verwendet die VPI-Bits zur Realisierung eines proprietären Flow-Control-Mechanismus (FLOWmaster). VCI=0 bis VCI=31 sind für Zwecke des CCITT bzw. des ATM-Forums reserviert.

Die unterstützten Wertebereiche für VPI bzw. VCI unterscheiden sich erheblich. So sind bei den Switches pro Interface jeweils maximal 12 Bits für VPI oder VCI konfigurierbar (Default: 4 Bits VPI, 8 Bits VCI). Die Endgeräte unterstützen allesamt nur VPI=0.

Die VCI/VPI-Kombination für eine spezielle Verbindung wird im Falle eines PVC auf allen beteiligten Geräten definiert. Im Falle eines SVC wird sie während der Signalisierungsphase bei Verbindungsaufbau vom Switch vorgegeben. Aufgrund der unterschiedlichen Wertebereiche und mangelhafter Dokumentation ergaben sich anfangs einige Probleme. Bei gescheiterten Verbindungsaufbauversuchen inkrementierte beispielsweise der Switch die VCI-Nummer auf dem Interface und nach Erreichen des per Default eingestellten Maximums von VCI=255 die VPI-Nummer, so daß von diesem Zeitpunkt an

keine Verbindung über das betreffende Interface mehr möglich war, da das Endgerät keine von Null verschiedenen VPI-Werte unterstützt. Durch Festlegung von VPI=0 (0 bits für VPI, LightStream100–Command: set interface) auf dem Switch konnte dieses Problem behoben werden. Der für die Belange des Testbetriebs ausreichende Default-Wert für die VCI-bits wurde beibehalten.

5.2 Classical IP / LAN Emulation (LANE)

Die Konfiguration von Permanent Virtual Circuits (PVC) war zwischen allen Komponenten problemlos möglich.

In einer ATM-SVC-Umgebung muß ein Mechanismus zur Auflösung von IP-Adressen in ATM-Adressen realisiert sein. Dies geschieht im allgemeinen durch Einsatz eines ATMARP-Servers, dessen ATM-Adresse auf den Clients konfiguriert werden muß. Die Adreß-Information der Clients wird beim Hochfahren ihres ATM-Interfaces automatisch per ATMARP beim Server registriert und danach in regelmäßigen Abständen vom Server aus aktualisiert. Vor dem Aufbau einer ATM-Verbindung zu einem anderen Client muß ein Host per ATMARP beim Server dessen ATM-Adresse erfragen. Dieser Vorgang bildet im verbindungsorientierten ATM-Netz das Äquivalent zum ARP-Broadcast klassischer LANs.

5.2.1 DEC

Unter den Betriebssytem Versionen V 3.2a/b/c werden die beiden folgenden Kommandos zur Konfiguration und Monitoring des ATM-Interfaces verwendet:

- `/sbin/atmconfig`
- `/usr/sbin/atmarp`

Mit dem *atmconfig* Kommando wird das ATM Subsystem konfiguriert. Nachfolgend die wichtigsten Optionen:

- **`up|down|status`** Herauf- und Herunterfahren des ATM-Interfaces, bzw. Abfragen des Status
- **`vclist [long]`** Auflisten der aktiven VCs
- **`+ pvc | - pvc | -vc`** Konfiguration von PVCs, bzw. Löschen von PVCs | SVCs
- **`+esi | -esi`** Konfiguration, bzw. Löschen von ATM Adressen
- **`source`** Ausführen der im `/etc/atm.conf` File spezifizierten Kommandos

Über das *atmarp* Kommando erfolgt die Konfiguration des ATMARP-Modules und der ATMARP-Tabelle sowie die Anzeige dieser Tabelle. Die wichtigsten Optionen sind:

- **`-a`** Anzeigen der kompletten ATMARP-Tabelle
- **`-d`** *`hostname`* Löschen des Eintags für *hostname* aus der ATMARP-Tabelle
- **`-h`** Anzeige, für welche Funktion (ATMARP Server/-Client) der lokale Host konfiguriert ist
  `-h server` Konfiguration des lokalen Hosts als ATMARP-Server
  `-h client` *`server_atm_addr server_ip_addr`* **`[`***`retry_time`***`]`** Konfiguration des lokalen Hosts als ATMARP-Client unter Angabe des ATMARP-Servers

- **-s** *atm_addr ip_addr* **[permanent]** erstellt in der ATMARP-Tabelle einen *atm_addr* Eintrag für die *ip-addr*

Mit dem Wechsel auf Digital UNIX 4.0 werden die Funktionaliät dieser beiden Kommandos erweitert und zusätzliche Befehle eingeführt.
So können nun mit *atmconfig* mehr Informationen angezeigt werden, z.B. über den Gerätetreiber oder das Signaling-Modul.
Da das ATM Interface jetzt für mehrere LISs (Logical IP Subnets) konfiguriert werden kann, müssen einige Optionen des *atmarp* um eine Subnetz-Indentifizierung erweitert werden.

Folgende Kommandos wurden hinzugefügt:

- `/usr/sbin/atmsig`
- `/usr/sbin/atmelan`
- `/usr/sbin/learp`

Der Befehl *atmsig* ermöglicht die Konfiguration und das Managemant des UNI Signaling-Moduls: z.B. Aktivieren und Deaktivieren des Signaling oder von ILMI auf einem ATM Interface.
atmelan und *learp* sind LAN Emulation (LANE) spezifische Kommandos: mit *atmelan* wird auf einem ATM Interface ein LANE Client (LEC) konfiguriert und überwacht, *learp* zeigt den Inhalt einer LANE Address Resolution Protocol Tabelle auf.

PVCs ließen sich sowohl zwischen DEC-Endgeräten, wie auch zu den Maschinen anderer Hersteller ohne Probleme aufbauen.

Für die Verwendung von SVCs muß jedes Endgerät eine ATM-Adresse bilden. Teile dieser Adresse erfragen die Maschinen über ILMI (Interim Local Managment Interface) vom Switch. Da der LS100 anfangs noch kein ILMI unterstützte, war es notwendig, auf den DEC-Maschinen durch Umsetzen einer Kernel-Variable (*atm_ilmi_pres*) das manuelle Konfigurieren der ATM-Adresse zu ermöglichen.
Danach konnten die DEC Maschinen sowohl als ATM ARP Client wie auch als ATM ARP Server gemäß RFC 1577 betrieben werden. Seitdem die LS100 ILMI unterstützen, kann auf die manuelle Konfiguration der ATM-Adresse verzichtet werden.
Der Einsatz von SVCs führte dann aber bei allen DEC Systemen in unregelmäßigen Abständen und ohne erkennbare Ursachen zu Systemabstürzen. Dieses Problem wurde bei DEC gemeldet, seine Behandlung zog sich jedoch in die Länge. Auch in der Zwischenzeit durchgeführte Betriebssystem-Upgrades auf den Maschinen (auf Digital UNIX V3.2c inklusive Firmware) und auf den LS100–Switches behoben diese Schwierigkeiten nicht.

Das DEC-Engeneering in den USA definierte als Ursache MTU-Unstimmigkeiten zwischen Maschine und ATM Switch. Da die geforderte Konfigurationsänderung auf dem Switch nicht möglich war, wurde der Fall mit dem Ergebnis *not supported* geschlossen. Eine daraufhin von DEC Deutschland eingeleitete Eskalation führte zu der Empfehlung zum Betriebssytem-Upgrade auf Version 4.0, die MTU–Konfigurationsmöglichkeiten auf DEC-Seite einschließt.
Seit dem Einsatz von Digital UNIX 4.0 auf den zwei Turbochannel-Maschinen sind keine entsprechenden Systemabstürze mehr aufgetreten. Dies führt zu der Annahme,

daß die Problemursache durch Änderungen im ATM Subsystem beim Release-Wechsel behoben wurde.

Mit dem neuen Betriebssystem-Release wird auch LANE (*LAN Emulation*) unterstützt, dessen Einsatz im nächsten Schritt getestet wird.

5.2.2 Sun

Die Konfiguration von Sun-ATM (Version 2.0) wird mit Hilfe folgender Dateien gesteuert:

- /etc/atmconfig (allgemein ATM)
- /etc/aarconfig (ATM Address Resolution / Classical IP)
- /etc/laneconfig (LANE)

Nach Änderung dieser Dateien muß Classical IP bzw. LANE neu initialisiert werden:

- /opt/SUNWatm/bin/aarsetup
- /opt/SUNWatm/bin/lanesetup

Bei der Initialisierung wird zunächst der ILMI-Daemon (/opt/SUNWatm/bin/ilmid) gestartet, um ATM-Adreßinformation auszutauschen. In den Konfigurationsdateien können bei Sun vordefinierte Variablen für die Bestandteile der Adresse verwendet werden.

Die Sun ATM Sbus-Karte unterstützt neben der ATMARP-Server- und -Client-Funktion auch statische IP/ATM-Adreßeinträge auf den Workstations, was natürlich in größeren Netzen Verlust an Flexibilität und einen erheblichen Verwaltungsaufwand bedeutet.

Im Verlauf der Tests kam es nach Konfigurationsänderungen gelegentlich vor, daß das ATM-Interface ohne ersichtlichen Grund 'hing' und mit den genannten Programmen nicht mehr zu initialisieren war. In einigen dieser Fälle war das Problem durch Reinitialisierung des ATM-Interfaces mit Hilfe des ifconfig-Befehls zu beseitigen (ifconfig sa0 down; ifconfig sa0 up). In anderen Fällen mußte die Maschine neu gestartet werden.

Die Sun-ATM-Software umfaßt eine ganze Reihe von Programmen, die die Untersuchung von Problemen bei der ATM-Konnektivität ermöglichen bzw. erleichtern:

- atmstat sa0 (Verbindungen, Eigenschaften und Statistik)
- aarstat sa0 (Status des ATM Address Resolvers)
- lanestat lane0 (LAN Emulation Status)
- qccstat sa0 (Status der Q.2931–Verbindungen)
- atmsnoop sa0 (Anzeige der ATM-Pakete auf dem Interface)

Leider fehlen Möglichkeiten, den Stand bestimmter Timer (Inactivity-Timer, Löschen von dynamisch im Kernel gespeicherter Adreß-Information) für spezielle Adressen oder Verbindungen zu überprüfen. Dies führte nach Konfigurations-Änderungen zu Unsicherheiten. Insbesondere fehlt eine Möglichkeit, die Kernel-Table per Command zu löschen bzw. Verbindungen zurückzusetzen.

Bei den Durchsatzmessungen per SVC trat zwischen zwei Sun-Rechnern das Problem auf, daß die Kommunikation unterbrochen wurde und danach nicht mehr neu aufgenommen werden konnte, weil die Informationen über das Netz bei beiden Rechnern nicht meht konsistent waren.

Diese Situation entstand, nachdem eine der beiden Sun's den SVC aus nicht näher zu ermittelnden Gründen abzubauen versuchte. Dabei löschten alle beteiligten ATM-Komponenten bis auf die zweite Sun die Information über diesen SVC aus ihren Tabellen. Wie eine Untersuchung mit dem ATM-Protokoll-Tester zeigte, versuchte die zweite Sun bei neuen Kommunikationsversuchen den alten, nur noch ihr bekannten SVC zu verwenden. Da sich Fehler dieser Art häuften, wurden die Durchsatzmessungen i.a. auf der Basis von PVC's durchgeführt.

5.2.3 IBM

Unter der urspünglich installierten Betriebssystemversion AIX 3.2.5 wurden lediglich PVCs für den Einsatz von Classical IP über ATM getestet. Damit konnten die IBM-Maschinen problemlos in das Testbett integriert werden, nachdem aufgrund der mangelhaften Dokumentation mit etwas Mühe der Bereich gültiger VCI-Werte bestimmt war.

Folgende Befehle dienen zur Konfiguration von Verbindungen und der Information:

- arp -t atm <options>: Anzeige bzw. Änderung der ATMARP-Tabelle. Optionen: -a: Anzeige, -d pvc: Löschen PVC, -s PVC/SVC Adreßeintrag kreieren
- atmstat atm0: Anzeige von Statistik-Angaben für das ATM-Interface. Optionen: -d (detailliert), -r (privilegiert: Reset der Statistik)
- atmsvcd: ?, in der Dokumentation nicht zu finden.

Die Einrichtung von SVCs unter der AIX-Version 4.0.1 erwies sich dagegen als sehr mühsam, was wiederum auf Schwächen der Dokumentation sowie auf Fehler in der Konfigurations-Oberfläche *smit* zurückzuführen war. Hinweise auf das geforderte Format bei der Eingabe von ATM-Adressen (Trennung der Bytes durch ":", "." oder keine Trennung?) fehlen und das Konfigurationsprogramm akzeptiert formal jedes beliebige Format. Bei Eingabe in einem falschen Format wird das zugrundeliegende Kommando ohne Fehlermeldung abgearbeitet, lediglich ein anschließendes Auflisten der eingegebenen Größen zeigt, daß die Eingabe falsch interpretiert wurde.

Ein weiterer Fehler zeigte sich beim Löschen eines vorher eingetragenen PVC. Wurde dieser korrekt, aber ohne Angabe der IP-Adresse des Zielsystems definiert, so konnte er anschließend mittels *smit* nicht mehr gelöscht werden, da das zugrundeliegende Kommando eine Option für die IP-Adresse mit einem Leerzeichen als Wert benötigt, *smit* aber aufgrund des fehlenden Eintrags diese Option ausläßt.

Nicht zuletzt wegen dieser Schwierigkeiten wurde die ATMARP-Server-Funktion auf RS6000 bisher nicht getestet.

Die Möglichkeiten zur Konfiguration des ATM-Interfaces und vor allem zur Abfrage von Information über konfigurierte Services und spezielle Verbindungen sind sehr begrenzt.

5.2.4 Cisco 7000

Eines der beiden ATM-Interfaces des Routers war während der Tests für *Classical IP over ATM* konfiguriert und übernahm die ATMARP-Server-Funktion.

Auf dem zweiten Interface wurde sowohl *Classical IP over ATM* wie auch LANE konfiguriert, ersteres für Performance-Tests über einen Router hinweg, letzteres für grundlegende LANE-Tests.

Der Router kann die Funktionen aller im LANE-Standard beschriebenen Server (LECS, LES, BUS) sowie die des Clients (LEC) übernehmen. Da die LS100–Switches und die LANE-fähigen SUNs die Server-Funktionen nicht unterstützen, wurden alle Server auf diesem ATM-Interface des Routers konfiguriert.

Spezielle Probleme bei Konfiguration oder Betrieb des Routers im ATM-Testbett traten nicht auf, als hilfreich erwiesen sich die vielfältigen Debbugging-Möglichkeiten.

5.2.5 LightStream 100 (LS100)

Laut Dokumentation zu den ersten drei Software-Versionen unterstützte der Switch die UNI–Schnittstelle (User Network Interface) in der Version 3.0. Obwohl ILMI (Interim Local Management Interface) ein Bestandteil der UNI 3.0–Spezifikation ist, wird es erst mit der 3er Version der Switch-Software voll unterstützt.

ILMI wird u.a. in einer SVC-Umgebung vom Endgerät benutzt, um vom Switch den ATM-Adreß-Prefix zu erfahren, der auf den beiden Switches jeweils mit dem 'set local'-Befehl definiert werden mußte. Dieser bildet zusammen mit der 6 Bytes langen MAC-Adresse des Host-Interfaces und einem zusätzlichen Selector-Byte (im allgemeinen X'00', bei DEC X'3A') eine eindeutige, 20 Bytes lange ATM-Adresse.

Andererseits übermittelt das Endgerät dabei dem Switch seine MAC-Adresse, die dort beim Aufbau der dynamischen, switch-internen Routing-Tabelle verwendet wird. Dies funktionierte bei den älteren Software-Versionen noch nicht. SVC-Routing-Einträge mußten auf den Switches für alle Zielsysteme explizit eingetragen werden. Mit der neuen Software müssen lediglich die Routing-Einträge für die Kommunikation zwischen den Switches manuell eingetragen werden.

Ab Software-Version 2.5 wurde auch IISP voll unterstützt (auch Soft PVC/PVP, s.u). Von diesem Zeitpunkt an mußte auf einem Switch/Switch-Link jeweils eine Seite als User- und die andere Seite als Network-Interface definiert werden, um SVC-Verbindungen über beide Switches hinweg zu ermöglichen (Lightstream-Command: set atmsig).

Außer der Eintragung der LECS-Adresse (LS100–Command: set configserver) sind auf den Switches keine LANE-spezifischen Aktionen erforderlich. Diese Eintragung wird von den LAN Emulation Clients (LEC) per ILMI abgefragt, allerdings wurde diese Funktion nur vom router-internen LEC unterstützt. Die Sun-LECs benutzen stattdessen die sogenannte 'wellknown address' des LECS (X"47007900000000000000000000:00A03E000001:00"). Da diese nicht mit der tatsächlichen LECS-Adresse im Testbett übereinstimmte (und demzufolge auch keine Routing-Einträge auf den Switches existierten), konnte eine LECS-Verbindung nicht automatisch aufgebaut werden. Die LECS-Adresse mußte auf den Sun-Rechnern manuell eingetragen werden (/etc/laneconfig).

Die Bedienung der Switch-Konsole ist ausgesprochen unkomfortabel. Mit den Befehlen müssen teilweise lange Listen numerischer Parameter übergeben werden, als Beispiel sei ein Kommando zitiert, mit dem man einen PVC (bidirektional, traffic type: unspecified bit rate, VPI/VCI 0/103) zwischen den Links 2 und 7 eines Switches definiert:

```
pvc est 1 2 2 0 103 512 0 0 7 0 103 512 0 0
```

Da kein Line Editing möglich ist, müssen schon abgesetzte Befehle im Fall von Tippfehlern komplett neu eingetippt werden. Das Laden von Konfigurationsdateien per tftp -

wie es von anderen Cisco-Produkten bekannt ist - ist beim LS100 nicht möglich, ebenso fehlt die Möglichkeit, per Kommando die komplette Switch-Konfiguration aufzulisten.

Es gibt praktisch keine Debugging-Möglichkeiten, lediglich einige Kommandos mit denen man sich Statistiken (show traffic) oder die per ILMI registrierten Endsysteme (show dynamicroute) anzeigen lassen kann.

In einigen Fällen war bei den frühen Software-Versionen nach Änderungen der Konfiguration im Rahmen der Tests ein Reset erforderlich, um die Maschine in einen definierten Zustand zu versetzten.

5.3 Soft permanent virtual connection / permanent virtual path (soft PVC/PVP)

Die Definition eines PVC zwischen zwei ATM-Hosts erfordert den Aufbau von PVCs auf jedem Link im ATM-Netz zwischen den beiden Rechnern. Dies bedeutet einen extremen Konfigurationsaufwand in großen ATM-Netzen.

Das von den Cisco Switches unterstützte Soft PVC/PVP ist ein Mechanismus, der Endgeräten PVCs zur Verfügung stellt – z.B. weil diese keine Signalisierung unterstützen –, dazu aber zwischen den Switches IISP-Prozeduren (Interim Inter Switch Protocol , manchmal bezeichnet als Private Network Node Interface (P-NNI) Phase 0) benutzt. IISP benutzt UNI-Signalisierungs-Prozeduren für die Switch/Switch-Kommunikation. Bei Verwendung von Soft PVC/PVP müssen nur die PVCs zwischen den Endgeräten und ihrem ATM-Switch definiert werden. Die Verbindung zwischen den beiden mit den Endgeräten verbundenen Switches (border switches) wird über IISP etabliert.

Obwohl in älteren Versionen vorgesehen, funktionierte IISP auf den Switches erst mit der Version 3. Da UNI-Signalisierung benutzt wird, muß auf einem Switch/Switch-Link jeweils eine Seite als User- und die andere Seite als Network-Interface definiert werden (Lightstream-Command: set atmsig)

In diesem Zusammenhang zeigten sich Unstimmigkeiten in der Cisco-Dokumentation: Laut Beschreibung steht am Beginn einer PVC/PVP-Verbindung der Eintrag einer 20–Bytes ATM-Adresse auf den Border Switches. Auch in der Beschreibung des 'set local'-Kommandos, mit dem diese Zuweisung geschehen soll, ist von einer 20–Bytes-Adresse die Rede, z.B. wird die Verwendung des 14. Bytes zur Angabe einer Port-Nummer beschrieben. An der Konsole können aber nur die 13 Bytes für den Adreß-Prefix eingegeben werden.

6 TCP(UDP)/IP-Performance über ATM

6.1 Einflußgrößen

Heutige Workstations sind an den Einsatz in traditionellen Netzwerken angepaßt. Die vom Hersteller vorgegebenen Einstellungen bestimmter TCI/IP-Parameter sind aber nicht geeignet, um beim Einsatz in ATM-Netzen mit hohen Bandbreiten optimale Performance zu erhalten. Hier sind Anpassungen erforderlich, die allerdings negative Auswirkungen auf die Performance bei Kommunikation über klassische Netze haben können.

Vor allem die folgenden Größen beeinflussen die Performance:

- Die Leistungsfähigkeit und Architektur der Endgeräte
- Größe des TCP 'sliding window', d.h. Anzahl der Bytes, die ein Sender transferieren darf, ohne eine Empfangsbestätigung des Empfängers abwarten zu müssen. Durch die Struktur des TCP-Headers ist diese Fenstergröße auf 64 kBytes beschränkt. Die DEC- und IBM-Endgeräte unterstützen die sog. Window-Scale-Option (siehe RFC 1323 [13]), die über einen Skalierungsfaktor auch größere Fenstergrößen erlaubt.
- Maximale Größe der TCP-Datensegmente (MSS, Maximum Segment Size). Die MSS wird beim Verbindungsaufbau vereinbart und innerhalb eines lokalen Netzes durch die zulässige Paketlänge auf dem Interface (MTU-size: Maximum Transfer Unit) bestimmt, indem hiervon 40 Bytes für die TCP/IP-Header abgezogen werden. Die MTU-Size muß innerhalb eines ATM-LIS (Logical IP Subnet) einheitlich sein, es wurde der in RFC1577 [11] empfohlene Wert von 9180 Bytes benutzt.
  Für Verbindungen mit Rechnern außerhalb des lokalen Netzes wird zur Vermeidung von Fragmentierung der vorgegebene Default-Wert von 536 Bytes für die MSS verwendet.
  Ein Mechanismus namens 'Path MTU Discovery' erlaubt es, dynamisch die größtmögliche MTU-Size für eine IP-Verbindung zu bestimmen. Die Sun-Systeme unterstützen diesen Mechanismus.
- Systempuffer (send/receive system buffer). Die Größe dieser Puffer wird über die Konfiguration der TCP/IP-Treiber auf dem jeweiligen Endsystem beeinflußt.
- Socketpuffer (send/receive socket buffer). Die Größe der Socketpuffer kann für eine bestimmte Anwendung jeweils durch setsockopt() definiert werden.
- Aufgrund der Segmentierung der Pakete in ATM-Zellen ist TCP empfindlich gegnüber Zellverlusten auf der ATM-Ebene. Bei Classical IP und einer MTU-size von 9180 Bytes wird ein TCP-Paket in 192 ATM-Zellen zerlegt. Geht eine davon bei der Übertragung verloren, so muß das komplette Paket erneut gesendet werden. Treten solche Zellverluste gehäuft oder in dem Sinne unglücklich auf, daß der Verlust weniger Zellen viele TCP-Pakete betrifft, sinkt die Performance unter Umständen drastisch ab.
- Durch die Beibehaltung der klassischen IP-Subnetzarchitektur müssen Endsysteme aus unterschiedlichen Subnetzen zur Zeit über Router kommunizieren, obwohl eine direkte ATM-Verbindung zwischen ihnen möglich wäre. Die Verwendung von Routern führt dabei zu unnötigem Protokolloverhead und zu Verzögerungen und damit auch zu Performanceeinbußen. Zeittransparenz und Skalierbarkeit gehen

verloren. Will man zwischen zwei Stationen direkte ATM-Verbindungen aufbauen, müssen sie so konfiguriert werden, daß sie in einem IP-Subnetz liegen.

6.2 Konfiguration der TCP(UDP)/IP-Treiber

Die Einstellung der Treiber-Parameter basiert auf Herstellerangaben bzw. auf Erfahrungswerten, die im RTB-NRW und bei eigenen Tests gewonnen wurden.

6.2.1 Sun

Die folgenden Parameter können mit dem Solaris-ndd-Kommando gesetzt werden. Die Syntax lautet (für das Beispiel tcp_mss_def):
`ndd –set /dev/tcp tcp_mss_def 9180`

Parameter	System Default	empfohlener Wert	Bedeutung
tcp_xmit_hiwat	8.192	65.535	TCP send buffer
tcp_recv_hiwat	8.192	65.535	TCP receive buffer
tcp_mss_def	536	9.140	TCP Default MSS
tcp_sth_rcv_hiwat	0	128.000	-
tcp_rwin_credit_pct	50	100	-
tcp_cwnd_max	32.768	65.535	congestion window size
tcp_naglim_def	4.095	2.048	nodelay
udp_xmit_hiwat	8.192	65.535	UDP send buffer
udp_recv_hiwat	8.192	65.535	UDP receive buffer
ip_path_mtu_discover	1	1	RFC 1191

Tabelle 1 TCP(UDP)/IP - Einstellungen Sun

6.2.2 IBM

Die folgenden Parameter können mit dem AIX-no-Kommando gesetzt werden. Die Syntax lautet (für das Beispiel sb_max):
`/usr/sbin/no -o sb_max=6000000`

Parameter	System Default	empfohlener Wert	Bedeutung
sb_max	65.536	6.000.000	Anzahl System buffer
tcp_sendspace	16.384	262.144	TCP send buffer
tcp_recvspace	16.384	524.288	TCP receive buffer
udp_sendspace	9.216	65.536	UDP send buffer
udp_recvspace	41.600	524.288	UDP receive buffer
rfc1323	0	1	window scaling option

Tabelle 2 TCP(UDP)/IP - Einstellungen IBM

6.2.3 DEC

Auf DEC OSF/1 / Digital UNIX Systemen kann der Kernel-Debugger benutzt werden, um Kernel Variablen zu lesen oder zu ändern (als root).
Der Kernel-Debugger wird über folgendes Kommando aktiviert:

```
dbx -k /vmunix
```

dbx Option	Funktion
`print` *kernel_var*	Anzeigen des Werts von *kernel_var*
`assign` *kernel_var* = *value*	Ändert den Wert von *kernel_var* auf *value*, bis zum nächsten Reboot
`patch` *kernel_var* = *value*	Ändert den Wert von *kernel_var* auf *value* dauerhaft (modifiziert `/vmunix`)

Ab Digital UNIX 4.0 gibt es die komfortablere Möglichkeit, über das XTool *Kernel Tuner* die Attribute des ladbaren Kernel-Subsystems zu pflegen (XDM-Login Manager: `dxkerneltuner`, CDE desktop: Application Manager —> System_Admin —> Monitoring Tuning —> Kernel Tuner). Die folgenden Parameter finden sich unter dem Subsytem *inet*.

Parameter	System Default	Einstellung	Bedeutung
tcp_sendspace	32768	65535	TCP send buffer
tcp_recvspace	32768	65535	TCP receive buffer
udp_sendspace	9216	9216	UDP send buffer
udp_recvspace	41600	41600	UDP receive buffer
tcp_mssdflt	536	4312	Default MSS
tcp_dont_winscale	0	0	window scale option (enabled)

Tabelle 3 TCP(UDP)/IP - Einstellungen DEC

6.2.4 Cisco 7000

Um auf einem Cisco 7000 die Paket-Transfer-Performance für IP-Verkehr zu verbessern, können je nach Hardware-Ausstattung zusätzliche Caches auf Interface-Basis benutzt werden. Bei gleichzeitiger Verwendung bestimmter Features der Router IP-Software (z.B. Accounting oder Access Listen) treten allerdings keine oder nur eingeschränkte Verbesserungen auf.

Bei Cisco 7000 Modellen mit einem einfachen *Switch Processor* sollte auf High-Speed-Interface Karten der Interface-Default *fast switching* auf *autonomous switching* geändert werden. Ist der Cisco 7000 mit einer *Silicon Switch Engine (SSE)* ausgestattet, sollte stattdessen *silicon switching* aktiviert werden (Kommando:`ip route-cache [cbus|sse]`).

6.3 Theoretische Durchsatzgrenzen

Die Übertragungsrate eines OC-3-Kanals beträgt 155.52 Mbps. Durch die Übertragung der ATM-Zellen in den SONET-Übertragungsrahmen sowie durch den gesamten Protokoll-Overhead ist die effektive Bandbreite, die bei Verwendung von TCP/IP für die Übertragung der Nutzdaten zur Verfügung steht, etwas geringer.

In der Classical-IP-Umgebung, bei einer MTU-Größe von 9180 Bytes, beträgt die maximale Nachrichtenlänge (MSS, Maximum Segment Size) 9140 Bytes, und der Overhead bildet sich wie folgt :

- ATM Layer Overhead 9.4%
- AAL 5 Overhead 0.3%
- LLC/SNAP-Verpackung (RFC1483) 0.09%
- IP Header 0.2%
- TCP Header 0.2%

Zur Übertragung einer 9140 Bytes langen Nachricht werden also 192 ATM-Zellen verwendet, und es werden insgesamt 192*53=10176 Bytes transportiert, was einen Overhead von 10.2% darstellt.

Dazu kommt noch der SONET-Transport und SONET-Path-Overhead, so daß von den 155.52 Mbps bei TCP/IP nur ungefähr 134 Mbps nutzbar sind. Bei der LAN-Emulation kommt noch ein zusätzlicher Overhead von 16 Bytes dazu, der durch die Verpackung der Ethernet-Pakete entsteht.

6.4 Test-Software

Für die Leistungsmessungen im ATM-Testbett wurde das frei verfügbare Netperf verwendet. Netperf wurde von Hewlett-Packard entwickelt und basiert auf dem Client/Server-Modell. Möglich sind Messungen des Durchsatzes sowie der Antwortzeiten (request/response) für verschiedene Protokolle, u.a. auch für TCP und UDP.

Bei den Messungen erlaubt Netperf, verschiedene Parameter zu definieren, wie z.B. bei TCP die Größe der Socket-Puffer (sowohl beim Sender als auch beim Empfänger), die Nachrichtengröße, die Menge der zu übertragenden Daten oder die Dauer des Tests sowie die Größe des Konfidenzintervalls. Dabei können die Parameter, die den Empfänger (den

Server) betreffen, von der Senderseite (Client) aus geändert werden. Dadurch lassen sich die Messungen durch entsprechende Scripts leicht automatisieren.

Wenn der Netperf-Client gestartet wird, baut er zunächst eine Kontrollverbindung zum Netperf-Server auf der entfernten Maschine auf. Diese Verbindung wird zur Übertragung von Konfigurationsinformationen und Ergebnissen von und zu der anderen Maschine benutzt. Es werden die TCP-Parameter übermittelt und danach die Socket-Optionen (vor allem Größe der Puffer) jeweils mit Hilfe von setsocopt() auf beiden Maschinen gesetzt. Anschließend baut Netperf zu dem Server eine zweite, für den jeweilgen Test passende Verbindung auf, die für die tatsächlichen Messungen benutzt wird.

Aus den zahlreichen Testmöglichkeiten wurden drei Arten von Tests gewählt: TCP-Stream-Test, UDP-Stream-Test und TCP-Request/Response-Test.

6.5 Ergebnisse der Messungen

Die TCP-Performance hängt von vielen Parametern ab. Dabei handelt sich vor allem, wie schon oben angedeutet, um die Größe der Puffer, die Größe der transportierten Pakete sowie um die gesamte Nachrichtenlänge der zu übertragenden Daten. Bei den Puffern wird aber noch weiter zwischen den Socket-Puffern und den Systempuffern unterschieden. Um die Abhängigkeiten und den Einfluß der Parameter auf die Kommunikationsleistung zu untersuchen, wurde eine Reihe von Tests durchgeführt, wobei jeweils ein oder zwei Parameter geändert wurden.

Zunächst wurde der Durchsatz in Abhängigkeit von der Größe der Socket-Puffer und der Nachrichtenlänge der zu übertragenden Daten gemessen. Die Socket-Puffer der sendenden und der empfangenden Maschine waren bei jeder Messung gleich groß. Für den Test kamen zwei gleichartige, schon einem Tuning unterzogene Sun-Maschinen zum Einsatz, bei denen die TCP-Systempuffer mit dem *ndd*-Kommando auf 64 kBytes eingestellt wurden. Die gemessenen Werte sind in Abb. 4 dargestellt und zeigen, daß die besten Durchsatzraten erst bei größerer Nachrichtenlänge sowie bei großen Socket-Puffern erreicht werden.

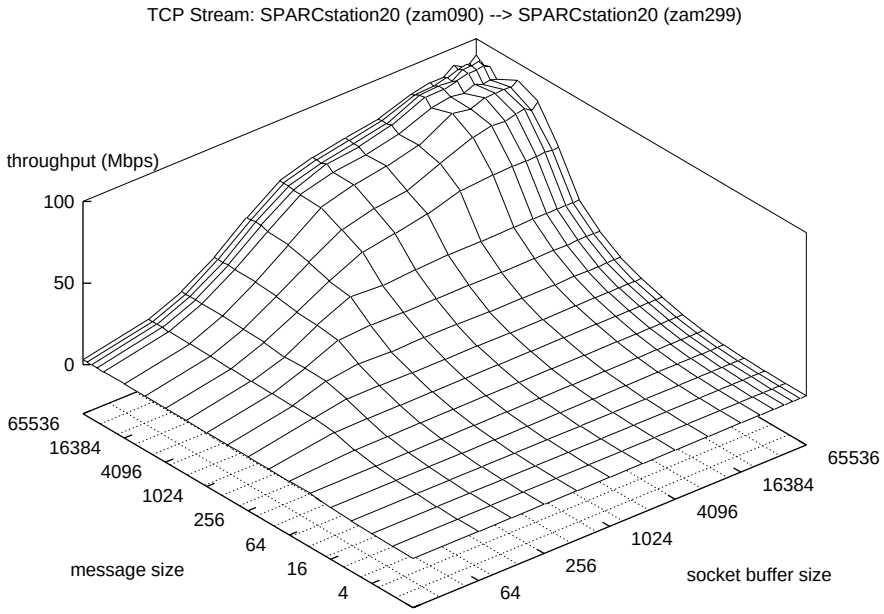

Abb. 4: Durchsatzrate SPARCstation20 → SPARCstation20
(receive/send buffer = 65535)

Ebenfalls auf den beiden Sun-Rechnern wurde der Einfluß der zwei Pufferarten (TCP-Systempuffer und Socket-Puffer) auf die Performance in zwei Meßreihen genauer untersucht.

Abb. 5 zeigt die Ergebnisse der ersten Messungen, bei denen zunächst die Größen der TCP-Systempuffer jeweils mit dem ndd-Kommando variiert wurden. Die Socket-Puffer wurden bei beiden Maschinen auf konstant 64 kBytes gesetzt. Die gewählte Nachrichtengröße betrug bei allen Messungen 9136 Bytes, so daß für jeden Datenblock nur ein Paket geschickt werden mußte. Es läßt sich hier deutlich erkennen, welch bedeutenden Einfluß die Größe der TCP-Systempuffer auf den Durchsatz hat.

Bei einer Default-Größe von 8192 Bytes sowohl beim Sender als auch beim Empfänger, ergab sich eine Datenrate von nur 42 MBps. Im Gegensatz dazu wurde bei den größtmöglichen Systempuffern von 64 kBytes auch eine größte Durchsatzrate von 85 MBps erreicht, was eine Steigerung der Performance um mehr als 100% darstellt. Außerdem kann man beobachten, daß die Größe des Empfangspuffers eine viel größere Wirkung auf die Performance hat als die Größe des Sendepuffers. Der Durchsatz steigt viel schneller mit wachsenden Empfangspuffer.

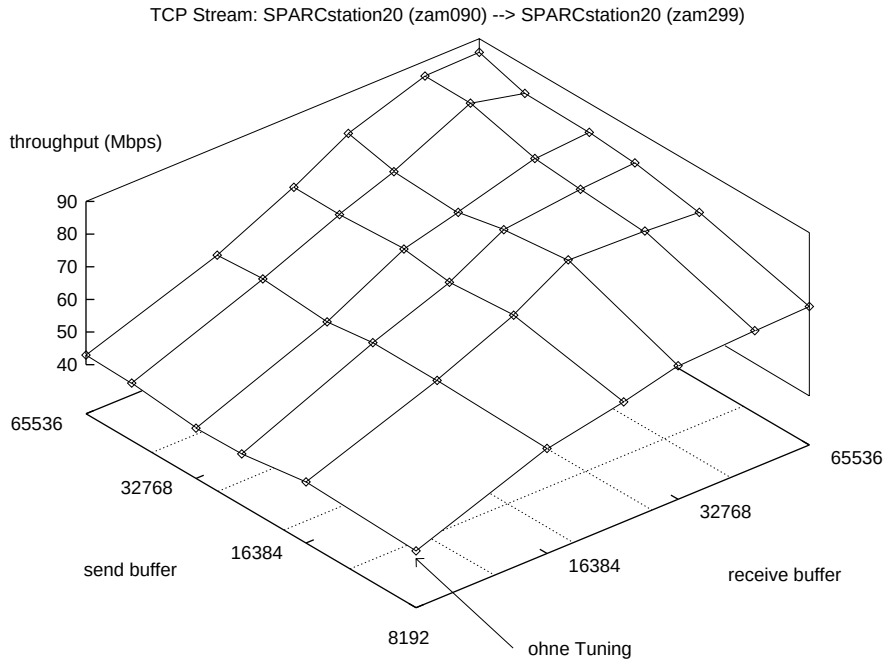

Abb. 5: Durchsatzrate bei verschiedenen Größen der Systempuffer (receive/send socket buffer = 65535, message length = 9136)

Der Einfluß der zweiten Pufferart (der Socket-Puffer) auf die Performance ist in Abb.6 dargestellt. Hier waren die TCP-Systempuffer auf beiden Maschinen gleich, stattdessen wurden die Socket-Puffer mit Netperf und setsockopt() geändert. Die Systempuffer wurden auf den maximalen (und wie die vorherigen Messungen zeigen auch bestmöglichen) Wert von 64 kBytes gesetzt. Die Nachrichtenlänge betrug bei allen Messungen wieder 9136 Bytes.

Die Meßergebnisse bestätigen die Vermutungen, daß für die Performance praktisch nur die Größe der Socket-Empfangspuffer eine Rolle spielt und der Einfluß der Größe der Socket-Sendepuffer in diesem Fall kaum spürbar ist. Die Größe des Sliding-Window wird vor allem durch die Größe der Socket-Empfangspuffer beeinflußt. Bei kleinerem Puffer wird das Fenster zu klein, um gute Resultate zu erzielen.

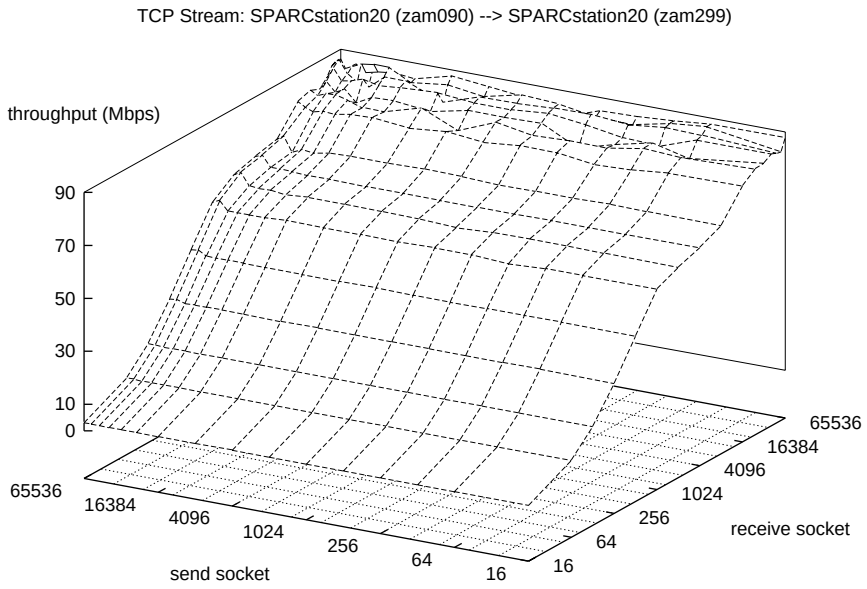

Abb. 6: Durchsatzrate bei verschiedenen Größen der Socket-puffer (receive/send buffer = 65535, message length = 9136)

Wie die zwei gerade beschriebenen Messungen zeigen, hängt der Durchsatz weniger von der Größe der Sendepuffer (Socket- oder Systempuffer), sondern vor allem von der Größe der Empfangspuffer ab. Es stellt sich die Frage, wie sich der Durchsatz ändert, wenn man beide Empfangspuffer bei fester Größe der Senderpuffer gleichzeitig variiert. Um dies zu untersuchen, wurde ein weiterer Test durchgeführt, dessen Ergebnisse in Abb. 7 dargestellt sind. Bei den Messungen waren sowohl der Socket- als auch der System-Sendepuffer auf 32 kBytes eingestellt, und die Länge der gesendeten Nachrichten betrug wieder 9136 Bytes. Das Ergebnis bestätigt die in Abb. 6 und 5 dargestellten Meßergebnisse und zeigt ähnliche Abhängigkeiten.

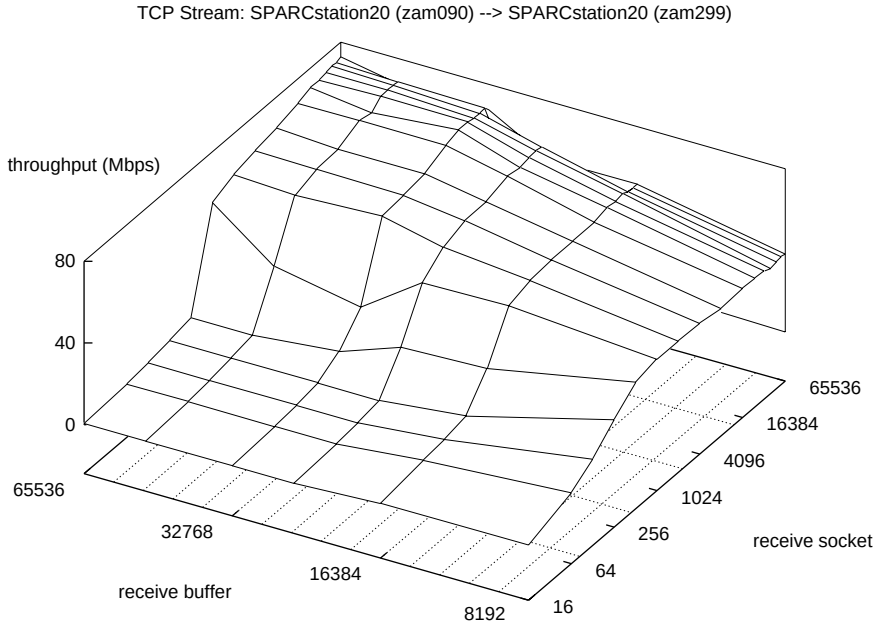

Abb. 7: Durchsatzrate in Abhängigkeit von der Größe des **receive buffer** (send buffer = 65535, message length = 9136)

Außer der Größe der Puffer hat auch die Nachrichtenlänge der zu übertragenden Daten einen großen Einfluß auf die Performance, was schon aus Abb. 4 ersichtlich ist. Aus diesem Grund wurden bei den nächsten Messungen die Durchsatzraten in Abhängigkeit von der Nachrichtenlänge untersucht. Abb. 8 zeigt die zunächst zwischen den beiden SUN-Maschinen gemessenen Werte. Die Messungen wurden jeweils bei drei verschiedenen Größen der Systempuffer durchgeführt, zuerst mit default-Werten (8KB), dann mit maximaler Größe (64KB) und schließlich noch mit einer Größe von 32KB. Zum Vergleich ist in das Abb. 8 auch der erreichbare Durchsatz über 10-Mbps-Ethernet aufgetragen. Die Socket-Puffer waren während aller Messungen gleich groß und wurden auf den maximalen Wert von 64 kBytes gesetzt.

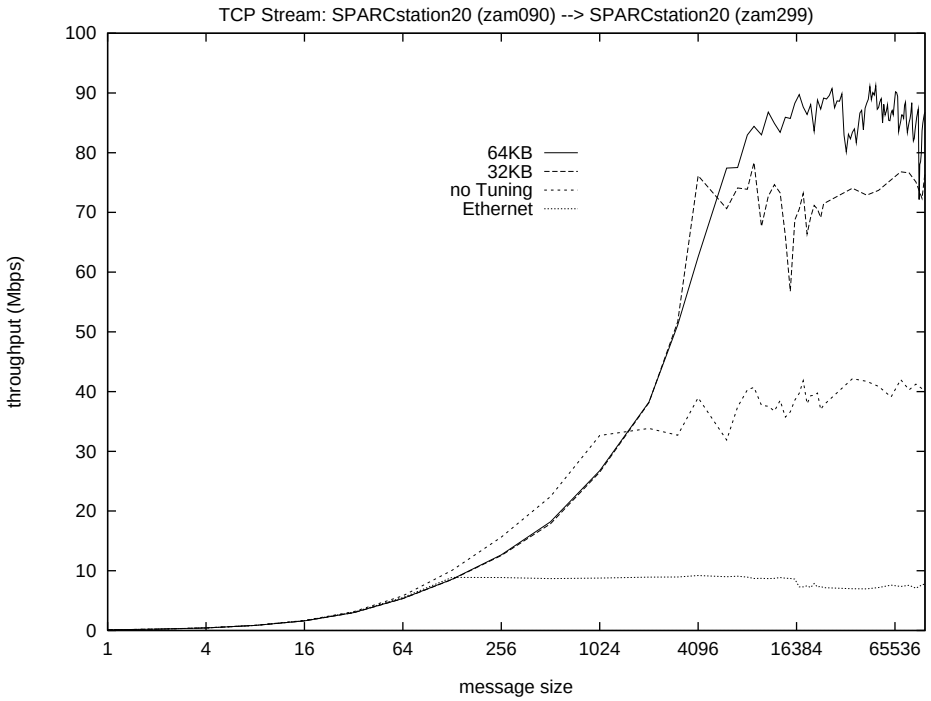

Abb. 8: Einfluß des Tuning auf den Durchsatz zwischen zam090 und zam299 (socket buffer = 65535)

Hier wird noch einmal der Unterschied in der Performance zwischen Maschinen mit und ohne Tuning deutlich. Wahrend das Tuning bei kleineren Nachrichtenlängen nicht so spürbar ist, verdoppelt sich die Performance bei größeren Längen der Nachrichten. Außerdem kann man den kontinuierlichen Anstieg des Durchsatzes bis zu einer bestimmten Nachrichtengröße beobachten. Danach bleibt der Durchsatz auf ungefähr gleichem Niveau, schwankt jedoch ein wenig von Nachrichtengröße zu Nachrichtengröße.

Um die Schwankungen genauer zu untersuchen, wurden im nächsten Test Durchsatzraten für Nachrichten gemessen, deren Länge im Bereich von 3500 bis 10000 Bytes jeweils in kleinen Schritten (4 Bytes) anstieg. Wie man Abb. 9 entnehmen kann, steigt der Durchsatz kontinuierlich an, solange die Länge der zu übertragenen Nachricht die maximale Segmentgröße nicht übersteigt, welche in diesem Fall 9140 Bytes beträgt. Dies läßt sich dadurch erklären, daß der Bearbeitungsaufwand in den Stationen weitgehend unabhängig von der Größe des Paketes ist. Die Performance steigt also an, solange nur ein Paket für jede Nachricht geschickt werden muß. Müssen zwei Pakete für jede Nachricht gesendet werden, wird der Kommunikationsaufwand größer und die Performance fällt um ca. 7.5 % ab. Abb. 9 zeigt auch sehr gut den Verlauf der Anstiegskurve, der Sägezähnen ähnelt. Dieses Phänomen, das auch schon in 24 beobachtet wurde, läßt sich auf die Implementierungsdetails bei der SUN-Maschine zurückführen.

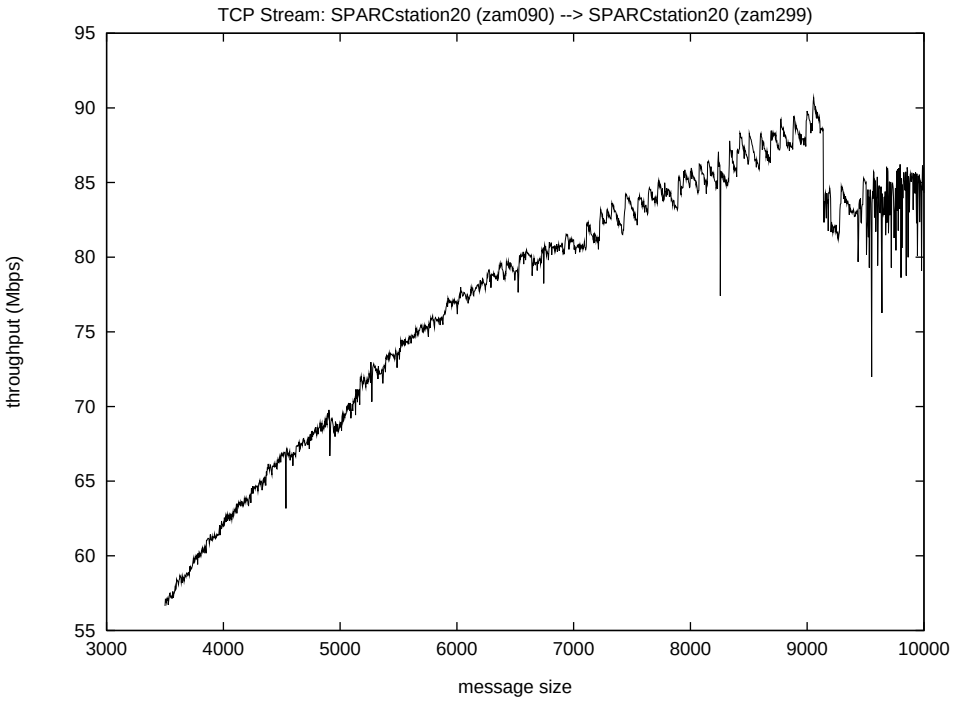

Abb. 9: Kontinuierlicher Anstieg der Performance im Detail

Vergleichbare Ergebnisse mit den oben beschriebenen ergaben sich auch bei ähnlichen Messungen mit den anderen Maschinen im Testbett. Dabei stellte sich allerdings die starke Abhängigkeit des erreichbaren Durchsatzes von der Leistungsfähigkeit der verwendeten Systeme heraus.

Um die Kommunikationsleistung der verschiedenen Maschinen im ATM-Testbett zu vergleichen, wurden in den folgenden Tests jeweils unterschiedliche Maschinen auf der Sender- und Empfangerseite eingesetzt.

In Abb. 10 wird zunächst die Sendeleistung der Maschinen verglichen. Als Empfänger wurde die stärkste Maschine, die zam004 (AlphaServer), eingesetzt, um die Zellverluste weitgehend zu vermeiden und die Sender dadurch nicht aufzuhalten. Die höchsten Werte von 116 Mbps bei einer Nachrichtenlänge von 100.000 Bytes erreichte die zam090 (SPARCstation20) als Sender. Die geringfügige Abweichung bei den für die identisch ausgestattete Sun zam299 gemessenen Werten erklären sich durch unterschiedliche ATM-Software-Versionen. Im gleichen Bereich (über 100 Mbps) liegen die Werte für die AlphaStation 200 (zam238). Die übrigen Maschinen erreichen nur Werten zwischen 60 und 70 Mbps. Auffällig ist, daß der Performance-Zuwachs im Bereich von Nachrichtenlängen bis zu 9140 Bytes bei den DEC-Maschinen deutlich schneller ansteigt als bei den Suns.

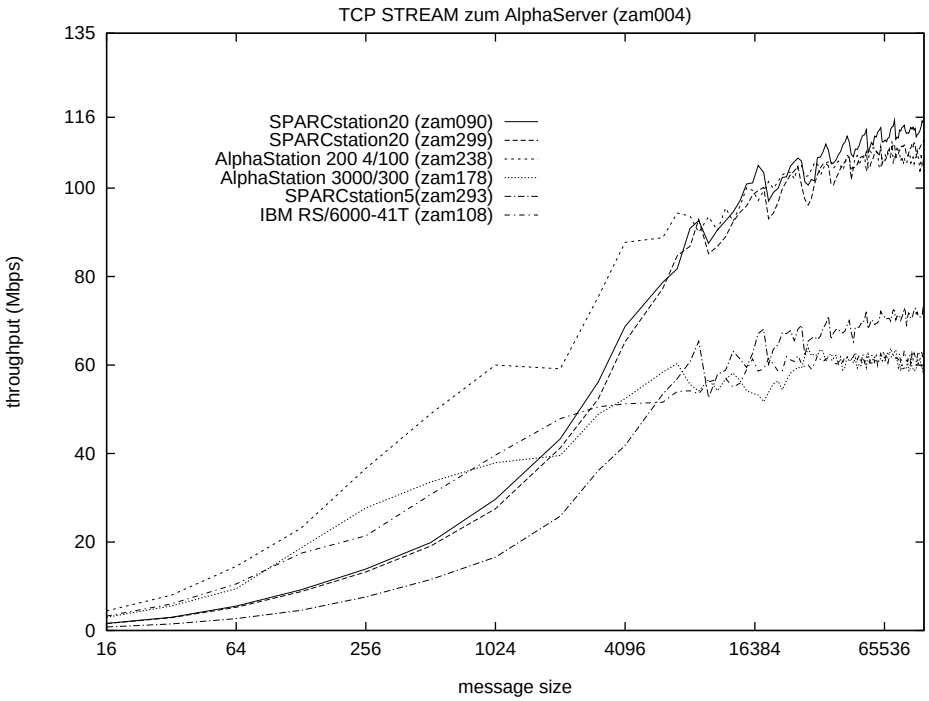

Abb. 10: Vergleich der Sendeleistung
(receive/send buffer = 65535, socket buffers = 65535)

Abb. 11 vergleicht dagegen die gleichen Maschinen in ihrer Empfangsleistung. Bei allen Messungen wurde ein relativ starker Sender, und zwar eine SPARC-Maschine (zam090), eingesetzt, die jeweils zusammen mit verschiedenen Empfängern getestet wurde. Während bei der Sendeleistung eine Aufteilung der Maschinen in zwei Gruppen zu erkennen ist, sind die entsprechenden Werte bei der Empfangsleistung stärker gestreut.

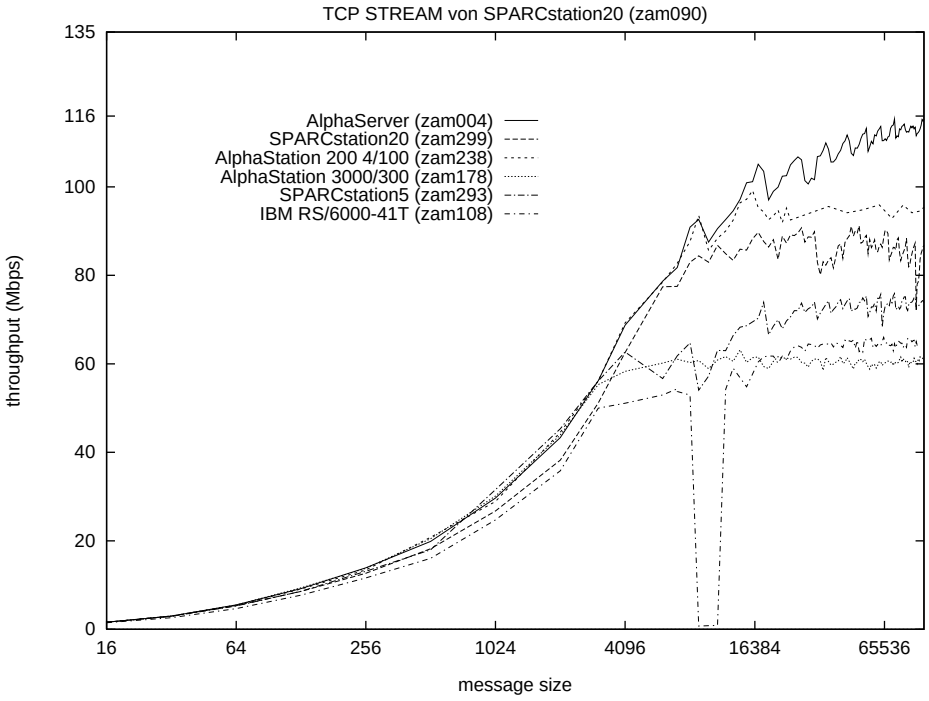

Abb. 11: Vergleich der Empfangsleistung
(receive/send buffer = 65535, socket buffers = 65535)

Bei diesen Tests fiel auf, daß die Software-Version beim Sun-Sender (1.0 bzw. 2.0) Einfluß auf den Verlauf der Performance-Kurve hatte. Während mit der alten Version die Messwerte für eine gegebene Nachrichtenlänge eine geringe Streuung aufwiesen, war die gemessene Performance bei der neuen Version (insbesondere bei Kommunikation mit einem Sun-Empfänger) von Messung zu Messung sehr instabil und fiel häufig auf Werte knapp über 10 Mbps ab.

Beim IBM-Empfänger bricht der Durchsatz aufgrund unglücklich auftretender Zellverluste für manche Nachrichtenlängen sogar ganz zusammen.

Nach den verschiedenen Tests zum Vergleich der Leistungsfähigkeit der im Testbett eingesetzten Maschinen wurden in weiteren Tests die Auswirkungen weiterer Einflußfaktoren auf die Performance untersucht.

Abb. 12 zeigt Performance-Unterschiede bei Verwendung der Scale-Option (Fenstergrößen von mehr als 64 kBytes, siehe 6.1) zwischen IBM/RS6000 und dem DEC-Server (Sun unterstützt diese Option nicht). Bei großen Nachrichtenlängen ergeben sich Performance-Gewinne in der Größenordnung von ca. 5 Mbps. Es ist zu vermuten, daß dieser Einfluß bei einer größeren round-trip-time, wie sie z.B. in Weitverkehrsnetzen auftritt, aufgrund des größeren bandwidth-delay-Produktes größere Bedeutung hat.

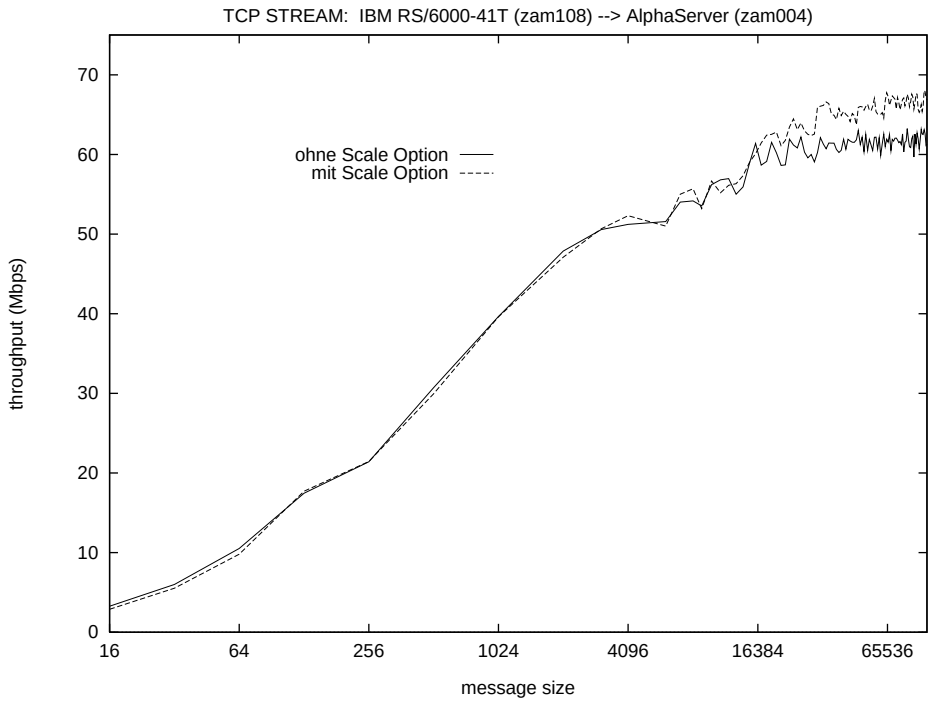

Abb. 12: Einfluß der Scale-Option (RFC 1323) auf die Performance

Mit Hilfe der NODELAY-Option (Nagel-Algorithmus) kann auf socket-Ebene gesteuert werden, ob kleine[1](nur teilweise gefüllte) Pakete abgeschickt werden oder nicht. Abb. 13 zeigt den Effekt dieser Option auf die Kommunikation zwischen den beiden Sun-Maschinen. Unterschiede sind natürlich nur für kleine Nachrichtenlängen (bis zum eingestellten Wert) erkennbar. Z.B. verringert sich der Durchsatz bei eingeschalteter NODELAY-Option und einer Message Size von 256 Bytes um die Hälfte.

[1] Bei Sun z.B. kann die Größe mit dem ndd-Kommando eingestellt werden (tcp_naglim_def-Parameter)

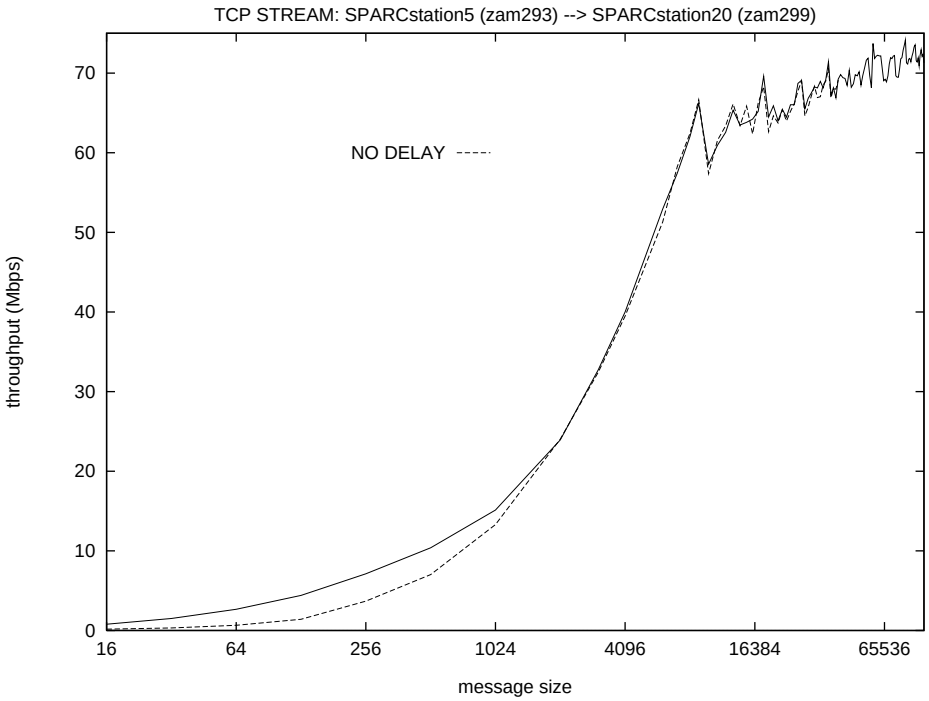

Abb. 13: Einfluß der NODELAY-Option auf die Performance
(receive/send buffer = 65535, socket buffers = 65535)

Als nächstes wurden zwei Maschinen getestet, die über einen Router kommunizieren, um dessen Einfluß sowie den Einfluß der Default-MSS-Größe auf die Performance zu untersuchen.

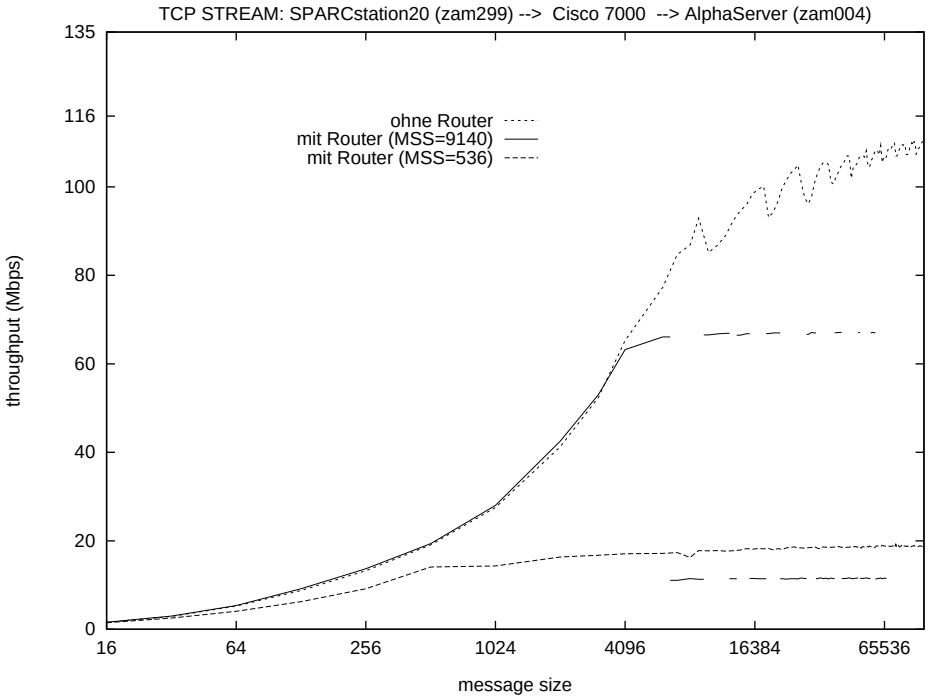

Abb. 14: Durchsatzraten bei der Kommunikation zwischen zwei Classical-IP-Netzen
(receive/send buffer = 65535, socket buffers = 65535)

Dazu wurde die Konfiguration des ATM-Testbetts kurzzeitig geändert. Es wurde auf demselben physikalischen ATM-Netz ein zweites Classical-IP-Netz gebildet. Es bestand

nur aus einer SUN-Maschine (zam299), die über den Router (zam002) mit dem AlphaServer im ursprünglichen Classical-IP-Netz kommuniziert. Beide Classical-IP-Netze wurden über verschiedene ATM-Interfaces an den Router angeschlossen. Es wurde zwischen einer SUN und dem AlphaServer (zam004) bei zwei verschiedenen MSS-Größen gemessen. 536 Bytes für die MSS werden als Default-Wert üblicherweise immer dann verwendet, wenn die beiden Endgeräte in unterschiedlichen IP-Netzen angesiedelt sind. Die dabei erzielten Durchsatzraten sind in Abb. 14 dargestellt, in dem zum Vergleich noch die Durchsatzrate aufgetragen ist, die zwischen diesen Maschinen bei direkter Verbindung (ohne Router) gemessen wurde. Der negative Einfluß des Routers auf den Durchsatz ist bei Nachrichtengrößen ab 4096 Bytes deutlich erkennbar.

Neben den Tests in den Classical-IP-Subnetzen wurde auch die Performance zwischen den beiden SUN-Maschinen in der LAN-Emulation-Umgebung untersucht. In Abb. 15 sind die Durchsatzraten zusammengefaßt, die zwischen den beiden SUNs bei verschiedensten Konfigurationen gemessen wurden. Man sieht, daß u.a. aufgrund der MTU-Größe von 1500 Bytes in dem emulierten LAN, nur Durchsatzraten bis etwa 22 Mbps erreichbar sind, was ca. 70% unter der Performance liegt, die zwischen diesen Maschinen in der Classical-IP-Umgebung (MTU-Size = 9180) gemessen wurde. Der Durchsatz liegt allerdings nur um ca. 25% unterhalb der Werte, die mit Classical IP bei einer MTU-Größe von 1500 Bytes erreicht werden, wobei sich diese Differenz aus dem LANE-Protokoll-Overhead ergibt.

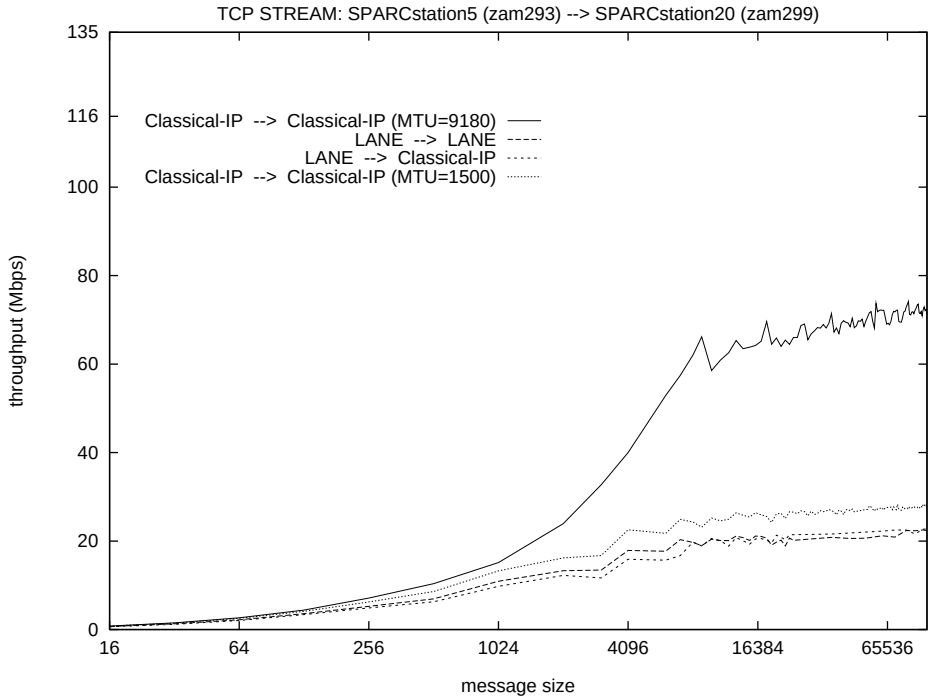

Abb. 15: Durchsatzraten bei Classical IP und LANE
(receive/send buffer = 65535)

Zum Schluß wurden noch zwei andere Testarten durchgeführt, und zwar UDP-Stream-Test und TCP-Request/Response-Test. Für die Messung der UDP-Performance wurde von der zam299 (SPARC20) zur zam178 (Alpha3000/300) gesendet. Es wurde jeweils die Übertragungsrate beim Sender und beim Empfänger gemessen. Alle Ergebnisse sind

noch zusammen mit den zwischen diesen Maschinen erzielten TCP-Durchsatzraten in Abb. 16 aufgetragen.

Man sieht, daß zuerst beide Durchsatzraten gleich sind und kontinuierlich ansteigen, weil keine Zell-/Paketverluste auftreten. Ab einer bestimmten Übertragungsrate ist jedoch der Empfänger nicht leistungsfähig genug, um die ankommenden Pakete schnell genug zu bearbeiten, und beginnt, Zellen zu verwerfen. Die Empfängerrate fällt dadurch beinahe linear mit wachsender Paketgröße. Die Senderate steigt jedoch weiter und nimmt erst wieder ab, wenn die UDP-Nachrichtenlange die MSS-Größe (9140 Bytes), d.h die Paketgröße, übersteigt. Sie fällt dabei sogar um 40% ab und liegt damit im Bereich, in dem noch keine Verluste beim Empfänger auftreten. Dadurch steigen beide Durchsatzraten mit wachsender Nachrichtenlänge noch einmal kurz zusammen an. Wird die Nachricht aber länger, dann sinkt die Empfängerrate wieder und fällt schließlich drastisch ab, sobald für eine UDP-Nachricht drei Pakete geschickt werden müssen. Wächst die Nachrichtenlänge noch weiter, dann steigt auch die Wahrscheinlichkeit, daß aufgrund eines einzigen Zellverlustes die ganze UDP-Nachricht verlorengeht, und der Durchsatz beim Empfänger bricht zusammen.

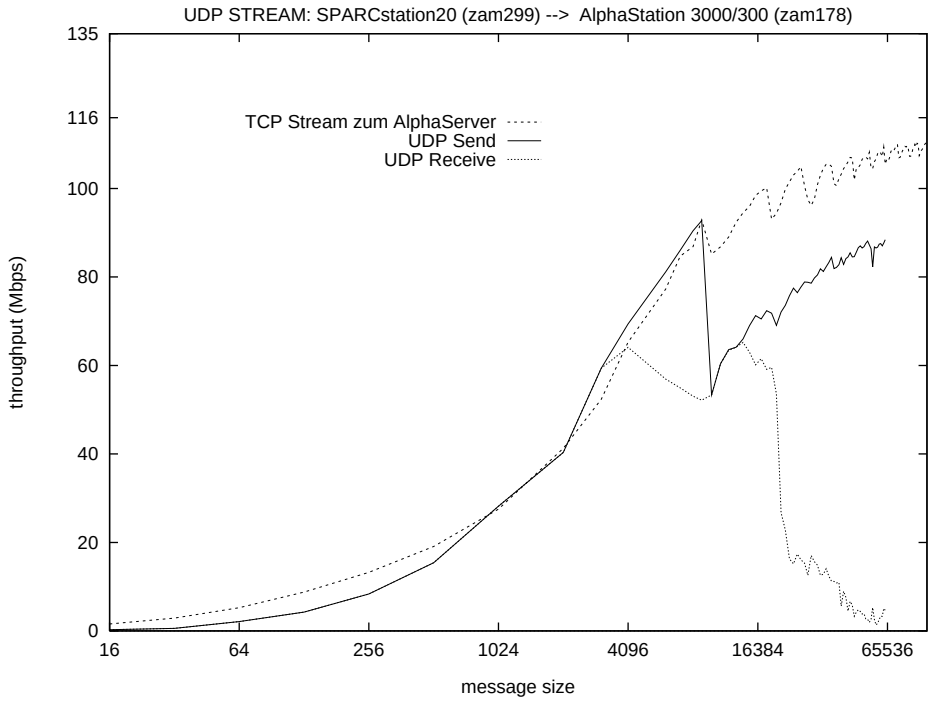

Abb. 16: UDP-Performance (receive/send buffer = 65535)

Die Ergebnisse des letzten Tests, bei dem das Antwortverhalten untersucht wurde, sind in Abb. 17 zu sehen. Es wurde die Anzahl der Transaktionen, die jeweils aus einem Request und einem Response bestehen, in Abhängigkeit von der Nachrichtenlänge und der Größe der Socket-Puffer pro Sekunde gemessen. Die besten Resultate wurden bei kleineren Nachrichtenlängen erzielt. Der maximale Wert von 800 Transaktionen/s läßt auf eine Round-Trip-Time von ca. 1,25 ms schließen.

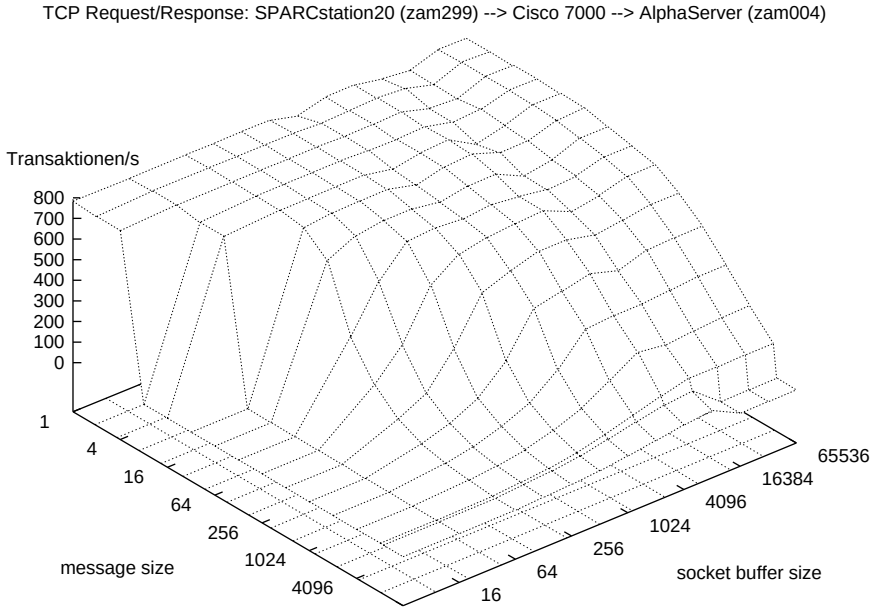

Abb. 17: TCP-Antwortverhalten

7 Zusammenfassung

Beim Aufbau des ATM-Testbetts im ZAM zeigte sich, daß zumindest die Verwendung von ATM-Komponenten unterschiedlicher Hersteller an vielen Stellen zu unvorhersehbaren Problemen führt. Zwar konnten viele dieser Probleme letztendlich gelöst werden, der dazu erforderliche Aufwand war jedoch recht hoch. Als besonders störend wurde empfunden, daß häufig im Grunde sehr einfache Fragestellungen, die in einer etablierten Technik innerhalb einiger Tage zu klären wären, zu (fast) unendlichen Geschichten eskalierten und letztlich einen unverhältnismäßig hohen Zeit- und Arbeitsaufwand erforderten.

Die Hauptgründe für die Probleme sind in der Tatsache zu sehen, daß die Hersteller vor dem Hintergrund der Entwicklungsprozesse, in denen sich die ATM-Standards heute zu einem großen Teil befinden, sehr unterschiedliche Implementierungen liefern und daher aus der Dokumentation nicht immer auf Interoperabilität geschlossen werden kann. Die Ursachen vieler Probleme wären ohne den Einsatz des Protokoll-Analyzers nur sehr schwer - wenn überhaupt - aufzuspüren gewesen.

Darüberhinaus erwies sich die Dokumentation in einigen Fällen als äußerst mangelhaft oder sogar fehlerhaft.

Als einzige Möglichkeit, einen über längere Zeit funktionierenden ATM-Betrieb mit allen beteiligten Geräten aufrechtzuerhalten, stellte sich Classical IP über ATM unter Verwendung von PVC-Verbindungen heraus, was gleichzeitig die Betriebsart mit dem höchsten Konfigurations- und Verwaltungsaufwand ist.

Um in einem ATM-Netz eine gute Performance zu erhalten, sind auf den für den Einsatz in klassischen Netzarchitekturen eingestellten Endgeräten Anpassungen erforderlich, die auf der anderen Seite die Kommunikation behindern können, wenn die Zielsysteme über klassische Netze erreicht werden. Als wichtigste Einflußgrößen stellten sich für den TCP-Verkehr die Nachrichtengröße und die Größen der System- und Socket-Puffer heraus, deren Werte in der Grundeinstellung aller Endgeräte auf einem für ATM-Kommunikation zu niedrigen Wert stehen.

8 Anhang

8.1 Glossar

AAL	ATM Adaptation Layer
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol
BUS	Broadcast and Unknown Server
IISP	Interim Inter-Switch Signalling Protocol
ILMI	Interim Local Management Interface
LANE	LAN Emulation
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LES	LAN Emulation Server
LIS	Logical IP Subnet
LLC	Logical Link Control
MAC	Medium Access Control
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
P-NNI	Private Network Node Interface
PVC	Permanent Virtual Circuit
PVP	Permanent Virtual Path
SNAP	SubNetwork Access Protocol
SONET	Synchronous Optical Network
SVC	Switched Virtual Circuit
UNI	User Network Interface
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier

8.2 Literatur

[1] D. Conrads: ATM — die Vermittlungs- und Multiplextechnik des Breitband-ISDN. Interner Bericht des Forschungszentrums Jülich, KFA-ZAM-IB-9504, März 1995.

[2] O. Kyas: ATM-Netzwerke. Aufbau — Funktion — Performance. DATACOM-Verlag 1995.

[3] A. Lukosek: Einsatz von ATM — Möglichkeiten und heutige Grenzen —. Berichte des Forschungszentrums Jülich; 3279, September 1996

[4] A. Alles: ATM Internetworking. CISCO Systems, May 1995

[5] D. Conrads: Die ATM-Technik und ihre Bedeutung für die KFA. Interner Bericht des Forschungszentrums Jülich, KFA-ZAM-IB-9629, Oktober 1996.

[6] D. E. Comer: Internetworking with TCP/IP, Volume 1, Principles, Protocols, and Architecture. Prentice Hall, 1991.

[7] D. E. Comer: Internetworking with TCP/IP, Volume 2. Design, Implementation, and Internals. Prentice Hall, 1991.

[8] D. Conrads: Datenkommunikation. Verfahren, Netze, Dienste. Vieweg Verlag, 1996.

[9] RFC 1755: M.Perez, F. Liaw, D. Grossman, A. Mankin, E. Hoffman & A. Malis. ATM Signaling Support for IP over ATM. February 1995.

[10] RFC 1626: R. Atkinson. Default IP MTU for use over ATM AAL5. May 1994.

[11] RFC 1577: M. Laubach. Classical IP over ATM. January 1994.

[12] RFC 1483: J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5. Finland, July 1993.

[13] RFC 1323: V.Jacobsen, R. Braden and D. Borman. TCP Extensions for High Performance. May 1992.

[14] RFC 1191: J. Mogul and S. Deering. Path MTU Discovery. November 1990.

[15] RFC 879: J.Postel. TCP maximum segment size and related topics. 1983.

[16] ATM Forum: User-Network Interface Specification Version 3.0. Prentice Hall, September 1993.

[17] ATM Forum: User-Network Interface Specification Version 3.1. Foster City, September 1994.

[18] ATM Forum: Interim Interswitch Signaling Protocol. Foster City, February 1995

[19] ATM Forum: LAN Emulation over ATM. Version 1.0. January 1995

[20] ATM Forum: LAN Emulation over ATM, Version 1.0 Addendum. December 1995

[21] Hewlett-Packard: Netperf: A Network Performance Benchmark. Revision 2.0. Information Networks Division, February 1995.

[22] R. Niederberger: Schnelle Netze für das Metacomputing. Anspruch und Wirklichkeit im RTB-NRW. Forschungszentrum Jülich (KFA), 1996

[23] F. Brockners, C. Cseh, M. Horneffer: IP über ATM Performance am Beispiel des RTB-NRW. RWTH Aachen, Universität zu Köln 1996

[24] T. Luckenbach, R. Ruppelt, F. Schulz: Performance Experiments within Local ATM Networks. GMD-FOCUS, June 1994.